UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/902,810	07/10/2001	Anders Hejlsberg	MS1-865US	6062

			EXAMINER
22801	7590	01/25/2006	RAMPURIA, SATISH

LEE & HAYES PLLC
421 W RIVERSIDE AVENUE SUITE 500
SPOKANE, WA 99201

ART UNIT	PAPER NUMBER
2191	

DATE MAILED: 01/25/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on *25 November 2005*.

2a) ☐ This action is **FINAL.**   2b) ☒ This action is non-final.

3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) *1,3,7,8,11-16,18-22,24,25 and 27* is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) ☒ Claim(s) *4,5,9 and 10* is/are allowed.

6) ☒ Claim(s) *1, 3, 7-8, 11-16, 18-22, 24, 25 and 27* is/are rejected.

7) ☒ Claim(s) *2,17,23 and 26* is/are objected to.

8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.

10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a) ☐ All   b) ☐ Some * c) ☐ None of:

      1. ☐ Certified copies of the priority documents have been received.

      2. ☐ Certified copies of the priority documents have been received in Application No. _____.

      3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *11/25/05*.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

1.   This action is in response to the RCE received on Nov. 25, 2005.

2.   The declaration filed on Set. 26, 2005 under 37 CFR 1.131 is sufficient to overcome the

Lam (.NET Framework Essentials, published in 2001) reference.

3.   Claims amended by the applicant: None.

4.   Claims allowed: 4, 5, 9 and 10.

5.   Claims Objected: 2, 17, 23 and 26.

6.   Claims pending in the application: 1, 3, 7-8, 11-16, 18-22, 24, 25 and 27.

7.   A request for continued examination under 37 CFR 1.114, including the fee set forth in

37 CFR 1.17(e), was filed in this application after final rejection. Since this application is

eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e)

has been timely paid, the finality of the previous Office action has been withdrawn pursuant to

37 CFR 1.114. Applicant's submission filed on Nov. 25, 2005 has been entered.

Information Disclosure Statement

8.   An initialed and dated copy of Applicant's IDS form 1449 filed on Nov. 25, 2005 isd

attached to the instant Office action.

Response to Arguments

9.   Applicant's arguments with respect to claims have been considered but are moot in view

of the new ground(s) of rejection.

Allowable Subject Matter

10.   Claims 4, 5, 9 and 10 are allowed.

The following is a statement of reasons for the indication of allowable subject matter:

The cited prior art taken alone or in combination fail to teach, in combination with the

other claimed limitations *an XmlReader constructor that enables initialization of a new instance*

of the XmlReader class; and a Read method that enables reading of nodes of the XML data via

the XmlReader class instance, as recited in claim 4, *XmlResolver class... a ResolveURI method*

that enables resolving the absolute URI from a base URI and a relative URI; and a GetEntity

method that enables mapping of the resolved URI to an object containing identified resource, as

recited in claim 9, and *XmlDataDocument class... a DataSet property that enables obtaining of a*

dataset that provides a relational representation of the data in a document; a Load method that

enables loading of the document using a specified data source and synchronizing the dataset

with the loaded data, as recited in claim 10. Claim 5 is depend on claim 4 therefore it is

allowable.

11.   Claim 2, 17, 23 and 26 objected to as being dependent upon a rejected base claim, but

would be allowable if rewritten in independent form including all of the limitations of the base

claim and any intervening claims. Claims 3, 18, 24 and 27 will be allowed because they are

directly or indirectly dependent on claims 2, 17, 23 and 26. If the issues with claim objection and

35 U.S.C. 101 are corrected as described below.

Claim Objections

12.   Claims 20, 21, 23, 24, 26 and 27 are objected to because of the following informalities:

Dependent claims 20 and 21, recites "A software architecture" in the preamble, it appears

to be a typo to support the independent claim 19, and it should have been recited as "A computer

system".

Dependent claims 23 and 24, recites "A software architecture" in the preamble, it

appears to be a typo to support the independent claim 22, and it should have been recited as "A

method". Appropriate correction is required.

Dependent claims 26 and 27, recites "A software architecture" in the preamble, it appears

to be a typo to support the independent claim 25, and it should have been recited as "A method".

Appropriate correction is required.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or
> any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and
> requirements of this title.

13.   Claims 1-3 and 16-18 are rejected under 35 U.S.C. 101 because the claimed invention is

directed to non-statutory subject matter.

Claim 1 and 16 is non-statutory because the language of the claim raises a question as to

whether the claim is directed merely to an abstract idea which would result in a practical

application producing a concrete, useful, and tangible result to form the basis of statutory subject

matter under 35 U.S.C. 101. Claim recites a software architecture for a distributed computing

system providing an application to configure request and an API, representing functional

descriptive material without a computer readable medium or computer implemented method, program per se are not tangibly embodied. Claims 2-3 and 17-18 are dependent on claim 1 and 16 respectively, therefore, further support the program without a computer readable medium or computer implemented method, program per se are not tangibly embodied thus amounts to only abstract idea and are nonstatutory.

Claim Rejections - 35 USC § 102

14.   The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless --
(a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

15.   Claim 6 is rejected under 35 U.S.C. 102(a) as being anticipated by published document by W3C (hereinafter called W3C) from Clark et al.

Per claim 6:

W3C discloses:

- an XPathNavigator constructor that enables initialization of a new instance of the XPathNavigator class (page 3 of 32, "section Introduction … XPath gets its name from its use of a path notation as in URLs for navigating through the hierarchical structure of an XML document");

- a MoveToFirst method that enables moving to a first sibling of a current node of XML data (page 3 of 32, "section Introduction XPath models an XML document as a tree of nodes");

- a MoveToNext method that enables moving to a next sibling of the current node (page 3 of 32, "section Introduction XPath models an XML document as a tree of nodes");

- a MoveToPrevious method that enables moving to a previous sibling of the current node (page 3 of 32, "section Introduction XPath models an XML document as a tree of nodes");

- a MoveToFirstChild method that enables moving to a first child of the current node (page 3 of 32, "section Introduction XPath models an XML document as a tree of nodes");

- a MoveToParent method that enables moving to a parent of the current node (page 3 of 32, "section Introduction XPath models an XML document as a tree of nodes"); and

- a NodeType property that enables obtaining the type of the node that is moved to (page 27 of 32, "section 5.3 Attribute Nodes Each element node has an associated set of attribute nodes; the element is the parent of each of these attribute nodes; however, an attribute node is not a child of its parent element").

16.   Claim 7 is rejected under 35 U.S.C. 102(a) as being anticipated by published document by W3C (hereinafter called W3C) from Clark et al.

Per claim 7:

W3C discloses:

- an XslTransform constructor that enables initialization of a new instance of the XslTransform class (page 4 of 90, "section Introduction A transformation expressed in XSLT describes rules for transforming a source tree into a result tree... A template is instantiated for a particular source element to create part of the result tree";

- a Load method that enables loading of the XSLT stylesheet (page 4 of 90, "section

  Introduction A transformation expressed in XSLT is called a stylesheet. This is because,

  in the case when XSLT is transforming into the XSL formatting vocabulary, the

  transformation functions as a stylesheet"); and

- a Transform method that enables transforming of the specified XML data using the

  loaded XSLT stylesheet and outputs the results (page 5 of 90, "section Introduction

  Transformations on XML documents that represent data are often of this kind (see [D.2

  Data Example]). XSLT allows a simplified syntax for such stylesheets").

17.   Claim 8 is rejected under 35 U.S.C. 102(a) as being anticipated by published document

by W3C (hereinafter called W3C) from Layman et al.

Per claim 8:

W3C discloses:

- a Schema class that contains a definition of a schema (page 2 of 34, "section

  Introduction... *Schemas* define the characteristics of classes of objects");

- a SchemaObject class that enables creating of an empty schema (page 5 of 34, "section

  Empty, Any, String, and Mixed Content... *Empty* and *any* content are expressed using

  predefined elements *empty* and *any*"); and

- a SchemaCollection class that contains a cache of defined XML Schema Definition

  language (XSD) and XML-Data Reduced Language (XDR) schemas (page 16 of 34,

  "section XML-Specific Elements... XML-Data schemas contain a number of facilities to

  match features of XML DTDs or to support certain characteristics of XML...").

18.   The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the

basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless --

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

19.   Claim 11 is rejected under 35 U.S.C. 102(e) as being anticipated by US Patent No.

6,209,124 to Vermeire et al. (hereinafter called Vermeire).

Per claim 11:

Vermeire discloses:

- an XmlWriter constructor that enables initialization of a new instance of the XmlWriter

    class (col. 20, lines 9-12 "It invokes the parser to convert the source code into a tree form,

    provide editing for the tree and generating the metadata XML using the XML Writer

    class"); and

- an WriteState property that enables obtaining of the state of an instance of the XmlWriter

    class (col. 15, lines 58-67 "This "state" information includes such things as the offset into

    the record, alignment requirements for the record, etc. The behaviors common to input

    and output record states were placed in the abstract class BaseRecordState. The

    InputRecordState and OutputRecordState extend this base class for behaviors that are

    specific for their I/O requirements."); and

- a plurality of Write methods that enable writing XML data via the instance of the

    XmlWriter class (col. 20, lines 9-12 "It invokes the parser to convert the source code into

a tree form, provide editing for the tree and generating the metadata XML using the XML

Writer class").

20.   Claims 12-15 and 21 is rejected under 35 U.S.C. 102(e) as being anticipated by US

Publication No. 2002/0169679 to Neumayer (hereinafter called Neumayer).

Per claim 12:

Neumayer discloses:

- An XmlValidatingReader class of an application program interface, embodied on one or

    more computer readable media, that enables DTD, XDR and XSD schema validation

    (page 2 and 3, paragraph 33 "the incoming data is validated... accomplished through a

    class called XMLValidator, which is a helper class to check if the XML data is valid"),

    the XmlValidatingReader class comprising:

- a ValidationType property that enables obtaining an indication of what type of validation

    to perform on a document (page 3 and 4, paragraph 33 "A method of validateXML can

    be used to check the given XML against the schema");

- a Read method that enables reading of nodes of the document so that validation of the

    document can be performed (page 3 and 4, paragraph 34 "an MLProcessor class, which

    extracts information from the XML and creates the rule object").

Per claim 13-15, 21:

The rejection of claim 12 is incorporated, and further, Neumayer discloses:

- an XmlValidatingReader class that enables DTD, XDR and XSD schema validation

  (page 2 and 3, paragraph 33 "the incoming data is validated... accomplished through a

  class called XMLValidator, which is a helper class to check if the XML data is valid").

Claim Rejections - 35 USC § 103

21.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

22.   Claims 1, 16, 19, 20, 22 and 25 rejected under 35 U.S.C. 103(a) as being unpatentable

over US Patent No. 6,418,448 to Sarkar (hereinafter called Sarkar) in view of US Patent No.

6,209,124 to Vermeire et al. (hereinafter called Vermeire) and further in view of Foody et al.

(hereinafter called Foody).

Per claim 1:

Sarkar discloses:

- A software architecture for a distributed computing system comprising: an application

  configured to handle requests submitted by remote devices over a network (col. 5, lines

  62-63 "transactions could be triggered through thin client windows communicating

  persistently with remote databases").

Sarkar does not explicitly disclose an application program interface to present functions used by

the application to access network and computing resources of the distributed computing system,

wherein the application program interface comprises a set of classes that make available

standards-based support for processing XML documents.

However, Vermeire discloses in an analogous computer system an application program

interface to present functions used by the application to access network and computing

resources of the distributed computing system (col. 10, lines 54-56 "the operator interface and

the application model which allows for actual processing of the particular language source

code"), wherein the application program interface comprises a set of classes that make available

standards-based support for processing XML documents (col. 20, lines 8-10 "invokes the parser

to convert the source code into a tree form, provide editing for the tree and generating the

metadata XML using the XML Writer class").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time

the invention was made to incorporate the method of using application program interface to

generate the code as taught by Vermeire into the method of receiving the data via remote

devices as taught by Sarkar. The modification would be obvious because of one of ordinary

skill in the art would be motivated to use an application program interface to provide the use of

the same data that is being used in the conventional system as suggested by Vermeire (col. 4,

lines 35-67).

Neither Sarkar nor Vermeire disclose wherein the set of classes are grouped in the application

program interface into a plurality of namespaces, and wherein a first of the plurality of

namespaces contains classes and enumerations to support XSLT (Extensible Stylesheet

Language Transformations) and a second of the plurality of namespaces contains an Xpath

parser and evaluation engine.

However, Foody discloses in an analogous computer system wherein the set of classes

are grouped in the application program interface into a plurality of namespaces (col. 10-11,

lines 65-66 and 1-14 "NameSpaces is the information to describe classes... Class Description

Framework is provided to enable this capability, and to enable OSAs to override built-in

functionality... Class Description Framework consists of a suite of classes which describe:

classes, instances, properties, functions (including methods), arguments, and exceptions..."),

and wherein a first of the plurality of namespaces contains classes and enumerations to support

XSLT (Extensible Stylesheet Language Transformations) and a second of the plurality of

namespaces contains an Xpath parser and evaluation engine (col. 11, lines 53-67

"...information... constructed... as NameSpaces (XSLT would be inherent in this process since

it consist of implementing the XML) are enumerated... would typically use subclasses of

each... classes... set information..." also see FIG. 2b and related discussion. Emphasis added).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time

the invention was made to incorporate the method of having plurality of classes into

namespaces used by application program interface as taught by Foody into the method of

receiving the data via remote devices as taught by the combination of Sarkar and Vermeire. The

modification would be obvious because of one of ordinary skill in the art would be motivated to

have plurality of classes into namespaces used by application program interface to provide a

single system enabling software objects from multiple heterogeneous object systems to

interoperate bi-directionally and to combine larger systems as suggested by Foody (col. 6, lines 52-64).

Per claim 16:

Sarkar discloses:

- one or more applications configured to be executed on one or more computing devices, the applications handling requests submitted from remote computing devices (col. 5, lines 62-63 "transactions could be triggered through thin client windows communicating persistently with remote databases").

Sarkar does not explicitly disclose a networking platform to support the one or more applications; and an application programming interface to interface the one or more applications with the networking platform, wherein the application program interface comprises a set of classes that make available standards-based support for processing documents written in a markup language.

However, Vermeire discloses in an analogous computer system a networking platform to support the one or more applications (col. 10, lines 54-56 "the operator interface and the application model which allows for actual processing of the particular language source code"); and an application programming interface to interface the one or more applications with the networking platform (col. 10, lines 54-56 "the operator interface and the application model which allows for actual processing of the particular language source code"), wherein the application program interface comprises a set of classes that make available standards-based

support for processing documents written in a markup language (col. 20, lines 8-10 "invokes the parser to convert the source code into a tree form, provide editing for the tree and generating the metadata XML using the XML Writer class").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of using application program interface to generate the code as taught by Vermeire into the method of receiving the data via remote devices as taught by Sarkar. The modification would be obvious because of one of ordinary skill in the art would be motivated to use an application program interface to provide the use of the same data that is being used in the conventional system as suggested by Vermeire (col. 4, lines 35-67).

Neither Sarkar nor Vermeire disclose wherein the set of classes are grouped in the application program interface into a plurality of namespaces, and wherein a first of the plurality of namespaces contains classes and enumerations to support XSLT (Extensible Stylesheet Language Transformations) and a second of the plurality of namespaces contains an Xpath parser and evaluation engine.

However, Foody discloses in an analogous computer system wherein the set of classes are grouped in the application program interface into a plurality of namespaces (col. 10-11, lines 65-66 and 1-14 "NameSpaces is the information to describe classes... Class Description Framework is provided to enable this capability, and to enable OSAs to override built-in functionality... Class Description Framework consists of a suite of classes which describe: classes, instances, properties, functions (including methods), arguments, and exceptions..."), and wherein a first of the plurality of namespaces contains classes and enumerations to support

XSLT (Extensible Stylesheet Language Transformations) and a second of the plurality of

namespaces contains an Xpath parser and evaluation engine, and a third of the plurality of

namespaces contains classes used to serialize objects into XML format documents or streams

(col. 11, lines 53-67 "…information… constructed… as NameSpaces (XSLT would be inherent

in this process since it consist of implementing the XML) are enumerated… would typically use

subclasses of each… classes… set information…" also see FIG. 2b and related discussion.

Emphasis added).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time

the invention was made to incorporate the method of having plurality of classes into

namespaces used by application program interface as taught by Foody into the method of

receiving the data via remote devices as taught by the combination of Sarkar and Vermeire. The

modification would be obvious because of one of ordinary skill in the art would be motivated to

have plurality of classes into namespaces used by application program interface to provide a

single system enabling software objects from multiple heterogeneous object systems to

interoperate bi-directionally and to combine larger systems as suggested by Foody (col. 6, lines

52-64).

Per claim 19 and 20:

Sarkar discloses:

- A computer system including one or more microprocessors and one or more software

    programs(col. 5, lines 62-63 "transactions could be triggered through thin client

    windows communicating persistently with remote databases"), the one or more software

programs utilizing an application program interface to request services from an

operating system(col. 5, lines 62-63 "transactions could be triggered through thin client

windows communicating persistently with remote databases").

Sarkar does not explicitly disclose the application program interface including separate

commands to request services that make available support for processing XML documents.

However, Vermeire discloses in an analogous computer system the application program

interface including separate commands to request services that make available support for

processing XML documents (col. 20, lines 8-10 "invokes the parser to convert the source code

into a tree form, provide editing for the tree and generating the metadata XML using the XML

Writer class").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time

the invention was made to incorporate the method of using application program interface to

generate the code as taught by Vermeire into the method of receiving the data via remote devices

as taught by Sarkar. The modification would be obvious because of one of ordinary skill in the

art would be motivated to use an application program interface to provide the use of the same

data that is being used in the conventional system as suggested by Vermeire (col. 4, lines 35-67).

Neither Sarkar nor Vermeire disclose the separate commands being grouped into different

namespaces a first namespace to support XSLT (Extensible Stylesheet Language

Transformations) and a second namespace to serialize objects into XML format documents or

streams.

However, Foody discloses in an analogous computer system the separate commands

being grouped into different namespaces a first namespace to support XSLT (Extensible

Stylesheet Language Transformations) (col. 10-11, lines 65-66 and 1-14 "NameSpaces is the

information to describe classes... Class Description Framework is provided to enable this

capability, and to enable OSAs to override built-in functionality... Class Description Framework

consists of a suite of classes which describe: classes, instances, properties, functions (including

methods), arguments, and exceptions...") and a second namespace to serialize objects into XML

format documents or streams (col. 11, lines 53-67 "...information... constructed... as

NameSpaces (XSLT would be inherent in this process since it consist of implementing the XML)

are enumerated... would typically use subclasses of each... classes... set information..." also see

FIG. 2b and related discussion. Emphasis added).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time

the invention was made to incorporate the method of having plurality of classes into

namespaces used by application program interface as taught by Foody into the method of

receiving the data via remote devices as taught by the combination of Sarkar and Vermeire. The

modification would be obvious because of one of ordinary skill in the art would be motivated to

have plurality of classes into namespaces used by application program interface to provide a

single system enabling software objects from multiple heterogeneous object systems to

interoperate bi-directionally and to combine larger systems as suggested by Foody (col. 6, lines

52-64).

Per claim 22:

Sarkar discloses:

- A method comprising: receiving one or more calls from one or more remote devices

     over a network (col. 5, lines 62-63 "transactions could be triggered through thin client

     windows communicating persistently with remote databases"),

performing the requested XML document processing (col. 5, lines 62-63 "transactions could be

triggered through thin client windows communicating persistently with remote databases").

Sarkar does not explicitly disclose wherein the one or more calls are to one or more functions

that make available support for processing XML documents; and performing the requested XML

document processing.

However, Vermeire discloses in an analogous computer system wherein the one or more

calls are to one or more functions that make available support for processing XML documents;

and performing the requested XML document processing (col. 20, lines 8-10 "invokes the parser

to convert the source code into a tree form, provide editing for the tree and generating the

metadata XML using the XML Writer class").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time

the invention was made to incorporate the method of using application program interface to

generate the code as taught by Vermeire into the method of receiving the data via remote devices

as taught by Sarkar. The modification would be obvious because of one of ordinary skill in the

art would be motivated to use an application program interface to provide the use of the same

data that is being used in the conventional system as suggested by Vermeire (col. 4, lines 35-67).

Neither Sarkar nor Vermeire disclose the one or more functions being grouped into a plurality of

namespaces with a first namespace containing an Xpath parser and evaluation engine and a

second namespace containing classes used to serialize objects into XML format documents or

streams.

However, Foody discloses in an analogous computer system disclose the one or more

functions being grouped into a plurality of namespaces with a first namespace containing an

Xpath parser and evaluation engine (col. 10-11, lines 65-66 and 1-14 "NameSpaces is the

information to describe classes... Class Description Framework is provided to enable this

capability, and to enable OSAs to override built-in functionality... Class Description

Framework consists of a suite of classes which describe: classes, instances, properties, functions

(including methods), arguments, and exceptions...") and a second namespace containing classes

used to serialize objects into XML format documents or streams (col. 11, lines 53-67

"...information... constructed... as NameSpaces (XSLT would be inherent in this process since

it consist of implementing the XML) are enumerated... would typically use subclasses of

each... classes... set information..." also see FIG. 2b and related discussion. Emphasis added).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time

the invention was made to incorporate the method of having plurality of classes into namespaces

used by application program interface as taught by Foody into the method of receiving the data

via remote devices as taught by the combination of Sarkar and Vermeire. The modification

would be obvious because of one of ordinary skill in the art would be motivated to have plurality

of classes into namespaces used by application program interface to provide a single system

enabling software objects from multiple heterogeneous object systems to interoperate bi-

directionally and to combine larger systems as suggested by Foody (col. 6, lines 52-64).

Per claim 25:

Sarkar discloses:

- A method comprising: calling, to one or more remote devices over a network (col. 5,

    lines 62-63 "transactions could be triggered through thin client windows communicating

    persistently with remote databases"),

- receiving, form the one or more remote devices, a response to the calling (col. 5, lines

    62-63 "transactions could be triggered through thin client windows communicating

    persistently with remote databases").

Sarkar does not explicitly disclose one or more functions that make available support for

processing XML documents; receiving, from the one or more remote devices, a response to the

calling.

However, Vermeire discloses in an analogous computer system one or more functions

that make available support for processing XML documents; receiving, from the one or more

remote devices, a response to the calling (col. 20, lines 8-10 "invokes the parser to convert the

source code into a tree form, provide editing for the tree and generating the metadata XML using

the XML Writer class").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time

the invention was made to incorporate the method of using application program interface to

generate the code as taught by Vermeire into the method of receiving the data via remote devices

as taught by Sarkar. The modification would be obvious because of one of ordinary skill in the

art would be motivated to use an application program interface to provide the use of the same

data that is being used in the conventional system as suggested by Vermeire (col. 4, lines 35-67).

Neither Sarkar nor Vermeire disclose the one or more functions being grouped into a plurality of

namespaces with a first namespace containing classes and enumerations to support XSLT

(Extensible Stylesheet Language Transformations) and a second namespace containing classes

used to serialize objects into XML format documents or streams.

However, Foody discloses in an analogous computer system the one or more functions

being grouped into a plurality of namespaces with a first namespace containing classes and

enumerations to support XSLT (Extensible Stylesheet Language Transformations) (col. 10-11,

lines 65-66 and 1-14 "NameSpaces is the information to describe classes... Class Description

Framework is provided to enable this capability, and to enable OSAs to override built-in

functionality... Class Description Framework consists of a suite of classes which describe:

classes, instances, properties, functions (including methods), arguments, and exceptions...")

and a second namespace containing classes used to serialize objects into XML format

documents or streams (col. 11, lines 53-67 "...information... constructed... as NameSpaces

(XSLT would be inherent in this process since it consist of implementing the XML) are

enumerated... would typically use subclasses of each... classes... set information..." also see

FIG. 2b and related discussion. Emphasis added).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time

the invention was made to incorporate the method of having plurality of classes into namespaces

used by application program interface as taught by Foody into the method of receiving the data

via remote devices as taught by the combination of Sarkar and Vermeire. The modification

would be obvious because of one of ordinary skill in the art would be motivated to have plurality

of classes into namespaces used by application program interface to provide a single system

enabling software objects from multiple heterogeneous object systems to interoperate bi-

directionally and to combine larger systems as suggested by Foody (col. 6, lines 52-64).

Conclusion

23.   The prior art made of record and not relied upon is considered pertinent to applicant's

disclosure.

Any inquiry concerning this communication or earlier communications from the

examiner should be directed to **Satish S. Rampuria** whose telephone number is **(571) 272-3732**.

The examiner can normally be reached on **8:30 am to 5:00 pm** Monday to Friday except every

other Friday and federal holidays. Any inquiry of a general nature or relating to the status of this

application should be directed to the **TC 2100 Group receptionist: 571-272-2100**

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, **Wei Y. Zhen** can be reached on **(571) 272-3708**. The fax phone number for the

organization where this application or proceeding is assigned is **571-273-8300**.

Information regarding the status of an application may be obtained from the Patent

Application Information Retrieval (PAIR) system. Status information for published applications

may be obtained from either Private PAIR or Public PAIR. Status information for unpublished

applications is available through Private PAIR only. For more information about the PAIR

system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR

system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Satish S. Rampuria
Patent Examiner/Software Engineer
Art Unit 2191
01/17/2006

WEI Y. ZHEN
PRIMARY EXAMINER